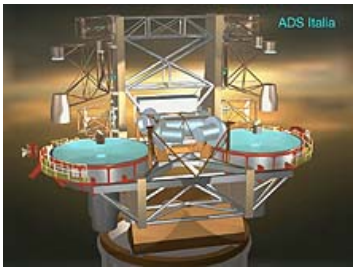



LBT PROJECT
2 x 8.4m TELESCOPE

Doc.No. : 675s001a
Issue : a
Date : 16-Feb-07

# LBT PROJECT

# 2 x 8.4m OPTICAL TELESCOPE

# General Purpose SERDES Communications Protocol Standard

| | Signature | Date |
|---|---|---|
| Prepared | J. Rosato | 16-March-07 |
| Reviewed | D. Ashby | 19-March-07 |
| Approved | | |

## 1. Revision History

| Issue | Date | Changes | Responsible |
|---|---|---|---|
| a | 16-Mar-07 | First draft | J. Rosato |
| | | | |
| | | | |
| | | | |

## 2. Table Of Contents

# 3. About this document

## 3.1. Purpose

The purpose of this document is to specify the communications protocol designed to serially transmit and receive data using the General Purpose SERDES Communications PC Board. Like the General Purpose SERDES Communications PC Board, the protocol itself is generic, that is, it is designed to allow the serialization of any data, so it can be used in many different applications.

To help clarify the use of this protocol, this document will provide examples of the use of this protocol with the General Purpose SERDES Communications PC Board in specific applications.

## 3.2. Reference Documents

[RD1] CAN Document 675s002a – General Purpose SERDES Communications PC Board Description and Operations Manual

## 3.3. Abbreviations

**SERDES -** Serializer-Deserializer

**MCSPU -** Mount Control System Processing Unit

**FPGA -** Field Programmable Gate Array

**GPSCOM-** General Purpose SERDES Communications

**CRC -** Cyclic Redundancy Check

**VHDL-** Very High Speed Integrated Circuit Hardware Description Language

**TBD -** To Be Determined

## 4. Serial Communication Function Example.

Figure 4.0.1 shows an example of a system where high-speed SERDES communications is used. Here status and control of telescope rotators can be accomplished remotely through a high-speed fiber link. The SERDES Communications protocol is implemented in the FPGA on the REEF board in the MCSPU and remotely on the General Purpose SERDES Communications PC Board [RD1]. The General Purpose SERDES Communications PC Board provides a standard hardware interface to allow the MCSPU to control hardware connected to the Rotator Control Interface Board.

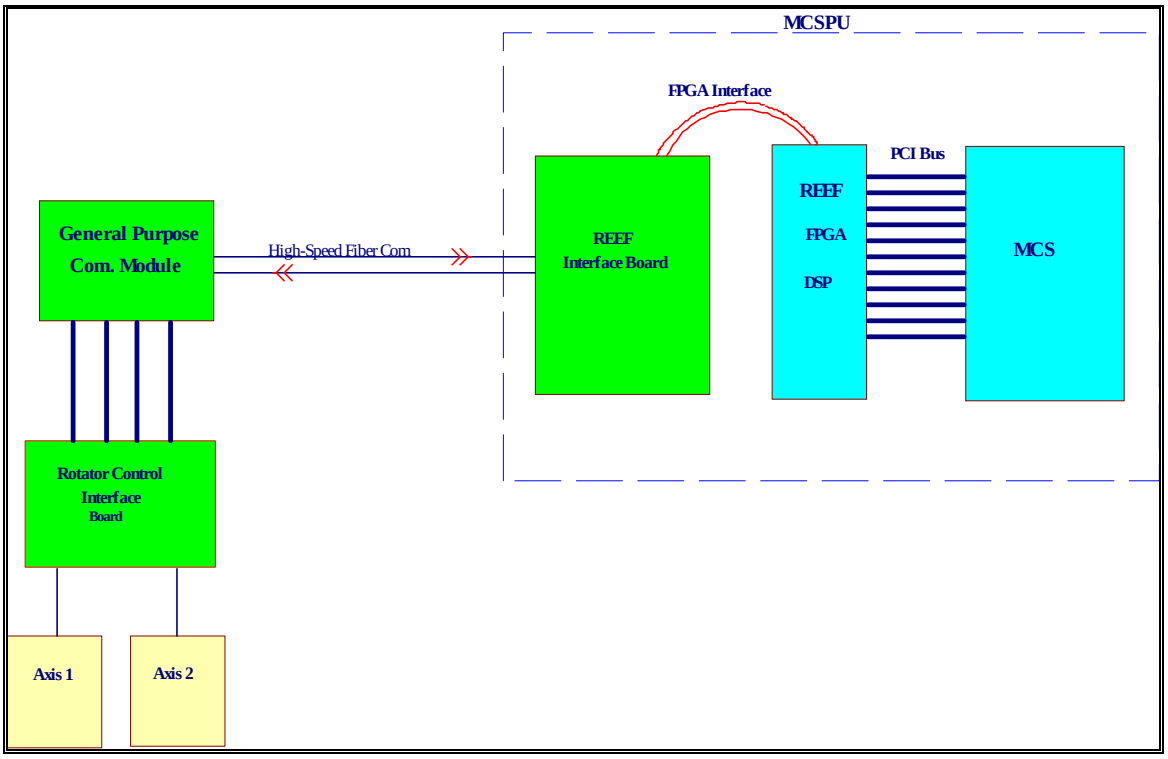



**Figure 4.0.2: Block Diagram of Rotator Distributed Control System**

## 5. Communications Protocol

The General Purpose SERDES Communications protocol defines a method to implement a bidirectional high-speed serial channel. This channel is designed specifically to serialize generic parallel data so that it can be transmitted and recovered serially.

The protocol defines high-speed data that is transmitted at the frame rate, as well as low-speed data that is time multiplexed and therefore updated at a rate slower than the frame rate.

This protocol defines the organization and use of frames of data. Specifically, it defines a "Data Out" frame and a "Data In" frame.

Because the direction of data is always relative to the position of the device in the system, the direction of data flow, in this document, is always described with respect to the General Purpose SERDES Communications PC Board output. Therefore, the "Data Out" frame is data sent from a remote system to the General Purpose SERDES Communications PC Board to be output.

To better understand the structure of the SERDES Communications Protocol, it is helpful to understand how it relates to specific hardware functions of the General Purpose SERDES Communications PC Board.

**General Purpose SERDES Communications PC Board**

The General Purpose SERDES Communications PC (GPSCOM) Board is a modular "daughter" card that plugs into an applications board.

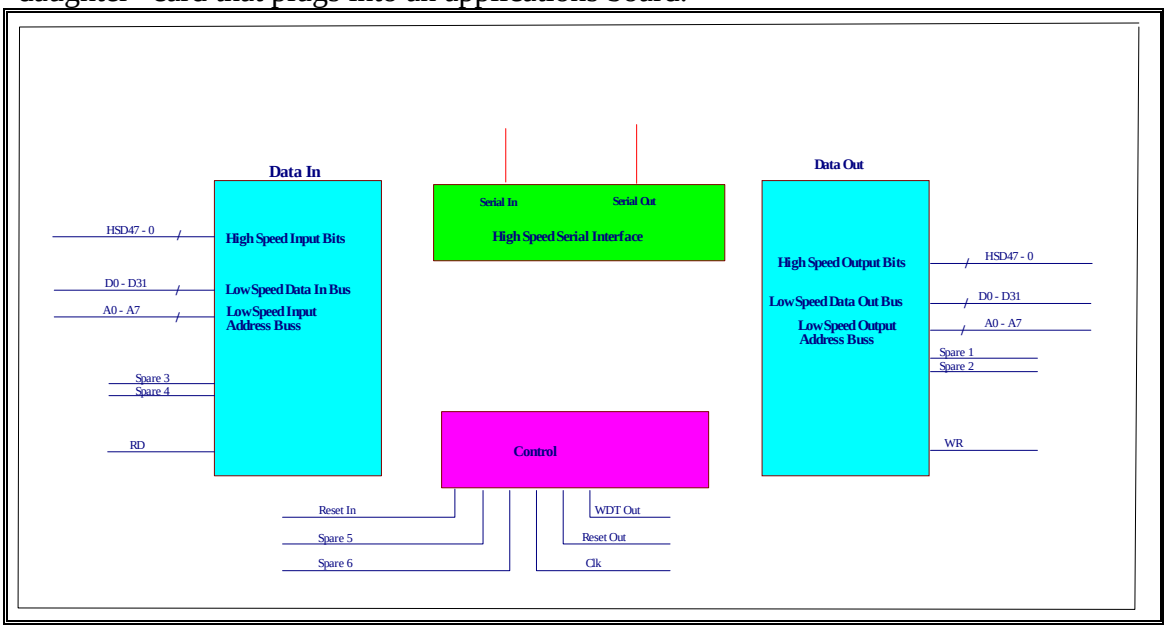



Figure 4.1.1

Figure 4.1.1 above shows a block diagram of the General Purpose SERDES Communications PC Board. There are four major blocks.

1. Data Out block, including high-speed data bus, and low-speed address, data and control signals.
2. Data In blocks, including high-speed data bus, and low-speed address, data and control signals.
3. High-Speed Serial Interface, which includes the Hotlink II chip and optical transceiver.
4. Control Block, with contains a clock output for synchronizing, reset in and out and a watchdog strobe output.

The structure of the General Purpose SERDES Communications Protocol is designed to allow control and synchronization of the Serial Interface, to provide Data for both the Data In and Data Out blocks, and a mechanism to provide both remote status and control of the General Purpose SERDES Communications PC Board (GPSCOM).

## 5.1. Data Out Block

The Data sent to the Data Out block is defined in the Data Out Frame.

**Data Out Frame definition**

| **Sync** | **SOF** | **High Speed Data** | **RXA** | **RX Data** | **CRC** | **EOF** |
|---|---|---|---|---|---|---|
| 8 bytes | 1 byte | 6 bytes | 1 byte | 4 bytes | 1 byte | 1 byte |

**Sync**- Sync characters are characters sent to fill the elasticity buffer and to maintain synchronization for the system operating from different clocks. This approach allows each Hotlink II device (local and remote) to run on separate clocks.

**SOF (Start of Frame) [1 byte] -** This byte defines the beginning of the "Data Out" frame of information.

**High Speed (Data) [6 bytes]** – These bits are associated with the high-speed data transmitted from the source to the remote GPSCOM Board. For the module these bits are general purpose. Each application will define these bits as appropriate.

The (GPSCOM) board's High-Speed output bits D47 – D0 are associated with the data in this field of the frame. The most significant bit of first byte in this field is mapped to D47. The 6 Byte field with associated GPCSOM data is show below.

| D47 | D39 | | | | D7 D0 |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte3 | Byte 4 | Byte 5 | Byte 6 |

**RXA – Receive Address [1 byte] –** The data in this field is output on the Low-Speed Output Address Buss. This byte is used to address devices that will accept Low-Speed Output Data. This field allows an interface connected to the GPSCOM board to write to a specific address on an application board.

See Section on Write Sequence timing for more information.

**NOTE:** Some of the addresses associated with the Low-Speed Data Bus program the behavior of the remote devices transmit sequencer. Therefore these addresses will not cause a write cycle to appear on the bus, as their destinations are registers inside the GPSCOM Board logic itself. These addresses are fixed in both their address location and in their bit definitions

**See section 5.2.1 Remote Sequencer Operation.**

**RX Data – Receive Data [4 bytes (32 bits)]** – This is the data associated with the address in the RXA field. This data is output on the Low-Speed Data Out Bus. The table below shows the relationship between these bytes and the Low-Speed Data Out Bus.

| D31 | | | D7 D0 |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte3 | Byte 4 |

**CRC (Cyclic Redundancy Check) –** This byte is calculated by the transmit process and appended to the frame. The receiving process must store an entire frame of data, and check for data integrity problems using the CRC byte sent. If this check finds an error in the data sent the entire packet of data is thrown away and the **Total Cyclic Redundancy Check error counter (CRC0)** and **Maximum Consecutive Cyclic Redundancy Check error counter (CRC1)** in the Status register are updated. See Status register definitions for more information.

**EOF (End of Frame) [1 byte] –**This byte indicates the end of a "Data Out" frame.

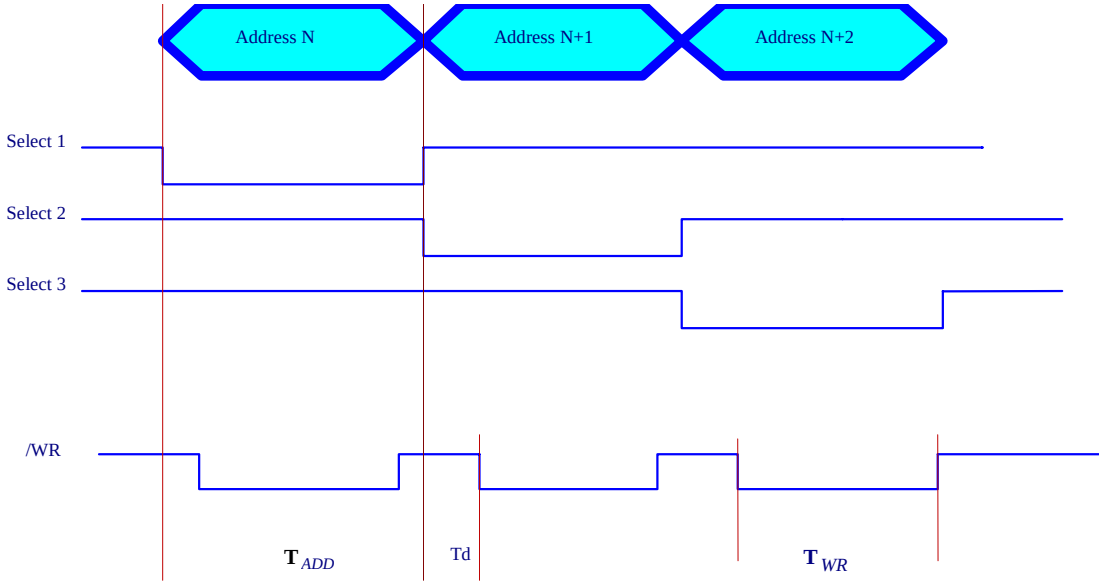
**Data Out Block Operation**

The FPGA on the GPSCOM board must take the data contained in the Data Out frame and do the following.

1. Store each byte of the frame while performing a CRC on the incoming data.
2. Compare the calculated CRC with the CRC sent from the transmission process.
   If the calculated and received CRC do not match, throw away the frame that was stored and update **Total Cyclic Redundancy Check error counter (CRC0)** and **Maximum Consecutive Cyclic Redundancy Check error counter. (CRC1)**
   See the CRC0 and CRC1 definitions in the Status register for more information.

3. If the calculated and received CRCs match, output the high speed data and produce a low speed write bus cycle with the address and data from the current frame.

Low Speed Data Write Timing




The diagram above shows a Low Speed Data Bus write transaction. Since the Data Bus and Address bus are separate buses, the Data and Address can be driven out on the bus simultaneously. The figure above shows only the Address Bus. Note that the select lines shown above represent the select line that would be produced by the application glue logic in response to the address on the bus. The /WR line can be use as both a bus transceiver enable (when it goes low) and a register clock. The application should latch the data into the device at the selected address, when the /WR line makes a low to high transition.

Assuming a 22 word frame as defined above, and a clock frequency of 50Mhz, the frame rate will be:

50Mhz/ word / 22 words/ frame = 2.272M/frames per sec = 440ns per frame.

$T_{ADD}$ = 440 ns

Assuming that the clock for the FPGA is 50 Mhz -> Td(min) = 1/50Mhz = 20nS

$T_d$ = 20 nS.

$T_{WR}$ = 400 ns.

### 5.1.1. The remote transmit process

**Special registers associated with the remote transmit process**

| | |
|---|---|
| **Control** | Used to program the remote transmitter sequencer |
| **Loop-backw** | Used to write data that is to be sent back to host for Self Test |
| | |
| | |

**Control**

| Bit 31 | Bit 30 | Bit 29 | Bit 28 | Bit 27 - 16 | Bit 15 – 8 | Bit 7 - 0 |
|---|---|---|---|---|---|---|
| **Heart beat** | **CRCR1** | **CRCR0** | **Afr** | **Unused** | **Stop address** | **Start address** |
| **1** | **1** | **1** | **1** | **12** | **8** | **8** |

The Control register defined above is mapped into the Low Speed Data address space and is written to by the process described in the Data Out section above. This register is 32 bits wide and exists at Address 0x0H.

**Start address**

The first 8 bits of the control register specify the start address for the sequencer.

**Stop address**

The second 8 bits of the control register specify the stop address for the sequencer. These addresses are output on the Low Speed Address Out bus to be decoded by external hardware.

Note, that Address 0 refers to the Control register on the GPSCOM board, so when the remote processes writes to this address, a bus cycle will not be created.

Additionally, the Loop-Back register is located at address 255 (0xFF). Accessing this address will also not cause a true bus cycle, because this information is stored in the internal logic of the Com Module.

**CRCR0**

Setting this bit will clear the **Total Cyclic Redundancy Check error counter** in the Status Register.

**CRCR1**

Setting this bit will clear the **Maximum Consecutive Cyclic Redundancy Check error counter** in the Status Register.

**Heart beat**

The state of the Heart beat bit in the control register is looped back in this bit. This signal can be used by the system software to insure that the serial channel is able to communicate in both directions. Additionally, this signal must toggle every 800 ms to insure the hardware watchdog timer on the GPSCOM board does not trigger.

The state of the hardware watchdog timer is output from the GPSCOM board to the attached application board. This signal should be monitored by the application and the hardware should be forced to a “safe state” when watchdog triggers.

## 5.2. Data In Block

While the data output by the GPSCOM board is controlled directly by the remote process, the data sent back from the GPSCOM board to the remote process is not under direct control. To allow some degree of control over the data sent back from the GPSCOM Board a programmable sequencer is required.

### 5.2.1. Remote Sequencer Operation-

The remote sequencer is implemented in the logic on the GPSCOM board. The sequencer produces read cycles on the Low Speed Data bus sequentially from the start address to the stop address. These cycles happen once per frame. The sequencer must also produce the read control signals to allow the hardware connected to the buss to interpret the address and present the data at the proper time.

**Read registers associated with the remote transmit process**

| | |
|---|---|
| **Status** | Displays status of communications channel |
| **Loop-backr** | This register contains a copy of the last data written to the Loop-backw register. |

## Status

| Bit 31 | Bit 30 | Bit 29 | Bit 28 | Bit 27 - 24 | Bit 23 | Bit 22 |
|---|---|---|---|---|---|---|
| **Heart beat** | **LFI** | **FTFI** | **AFI** | **Unused** | **AFC7** | **AFC6** |

| Bit 21 | Bit 20 | Bit 19 | Bit 18 | Bit 17 | Bit16 | Bits 15-8 | Bits 7-0 |
|---|---|---|---|---|---|---|---|
| **AFC5** | **AFC4** | **AFC3** | **AFC2** | **AFC1** | **AFC0** | **CRC1** | **CRC0** |

**Heart beat**

The state of the Heart beat bit in the control register is looped back in this bit. This signal can be used by the system software to insure that the serial channel is able to communicate in both directions.

**LFI- Link Fault Indication**

This bit is set to indicate one of four errors detected by the SerDes chip. These error are:

- Received serial data frequency outside of expected range

- Analog amplitude below expected levels
- Transition density lower than expected
- Receive channel disabled

**FTFI - Fiber Transmission Fault Indicator**

This signal returns the state of the fiber loss signal provided by the fiber optic transceiver. This signal is set to indicate that the optical signal from the fiber is no longer detected by the transceiver.

**AFI – Application Fault Indicator**

This is a general fault bit that is used to indicate an application fault. This bit is set if the remote monitor hardware detects a fault that requires the remote side application's attention. Details of the fault can be communicated using the AFC bits (AF0 – AFC7) located in the status register.

**AFC7-AFC0 – Application fault fields**

These bits are general purpose fault bits that are associated with specific hardware inputs from an application board. The meaning of these bits is application specific.

**CRC0 – Total Cyclic Redundancy Check error counter**

This count will be incremented each time a CRC error is detected until a count of 0xFF is reached. The count will not roll over. Setting the **CRCR0** bit in the control register resets this counter

**CRC1 – Maximum Consecutive Cyclic Redundancy Check error counter**

This count will indicate the maximum number of consecutive CRC errors. The count will saturate at 0xff and will not roll over to 0x00. Setting the **CRCR1** bit in the control register resets this counter

### 5.2.2. Data In Frame definition

| Sync | SOF | High Speed Data | TxA | Tx Data | CRC | EOF |
|---|---|---|---|---|---|---|
| 8 bytes | 1 byte | 6 bytes | 1 byte | 4 bytes | 1 byte | 1 byte |

**Sync**- These are characters sent to fill the elasticity buffer and to maintain synchronization for the system operating from different clocks. This approach allows each Hotlink II device (local and remote) to run on a separate clock and still remain synchronized.

**SOF (Start of Frame) [1 byte] -** This byte defines the beginning of the "Data In" frame of information.

**High Speed (Data) [6 bytes]** – These bits are associated with the high-speed data transmitted from the GPSCOM board to the remote receive process. For the module these are general purpose. Each application will define these bits as appropriate.

The (GPSCOM) board's High-Speed input bits D47 – D0 are associated with the data in this field of the frame. The most significant bit of first byte in this field is mapped to D47. The 6 Byte field with associated GPCSOM data is show below.

| D47 | D39 | | | | D7 | D0 |
|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte3 | Byte 4 | Byte 5 | Byte 6 | |

**TXA – Transmit Address [1 byte] –** This byte represents the address of the device on the Low Speed Bus where data is being read. This is the address produced by the sequencer in response to the setting in the control register.

**NOTE:** Some of the addresses associated with the Low-Speed Data Bus are used for monitoring the operation of the interface. These addresses are fixed in both their address location and in their bit definitions

**TX Data – Transmit Data [4 bytes (32 bits)]** – This is the data associated with the address in the TXA field. This is the data that was present on the Low-Speed Data In Bus at the time that the /RD signal made a low to high transition. See section below on read sequence timing

The table below shows the relationship between these bytes and the Low-Speed Data In Bus.

| D31 | | | D7 | D0 |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte3 | Byte 4 | |

**CRC (Cyclic Redundancy Check) –** This byte is calculated by the GPSCOM board transmit process and appended to the frame. The receiving process must store an entire frame of data, and check for data integrity problems using the CRC byte sent. If this check finds and error in the data sent the entire packet of data is thrown away and the CRC error bit in the Communication Status register is set.

**EOF (End of Frame) [1 byte] –** This byte indicates the end of a "Data in" frame.

**Data In Block Operation**

The FPGA on the GPSCOM board must take the data from the high speed inputs and the Low Speed Data bus and produce the Data In frame. To do this it does the following.

1. Starts the process for calculating a CRC for data sent.
2. Produces Sync and the SOF characters for the first part of the frame.
3. Produces a Low Speed Data bus cycle.
4. Places the high speed input states into the High Speed Data portion of the frame.
5. Places address of the last Low Speed Data bus cycle in the frame.
6. Places the data from the last Low Speed Data bus cycle in the frame.
7. Completes the calculation of the CRC and place it in the frame.
8. Places EOF character in the Frame.

Read Sequence Timing

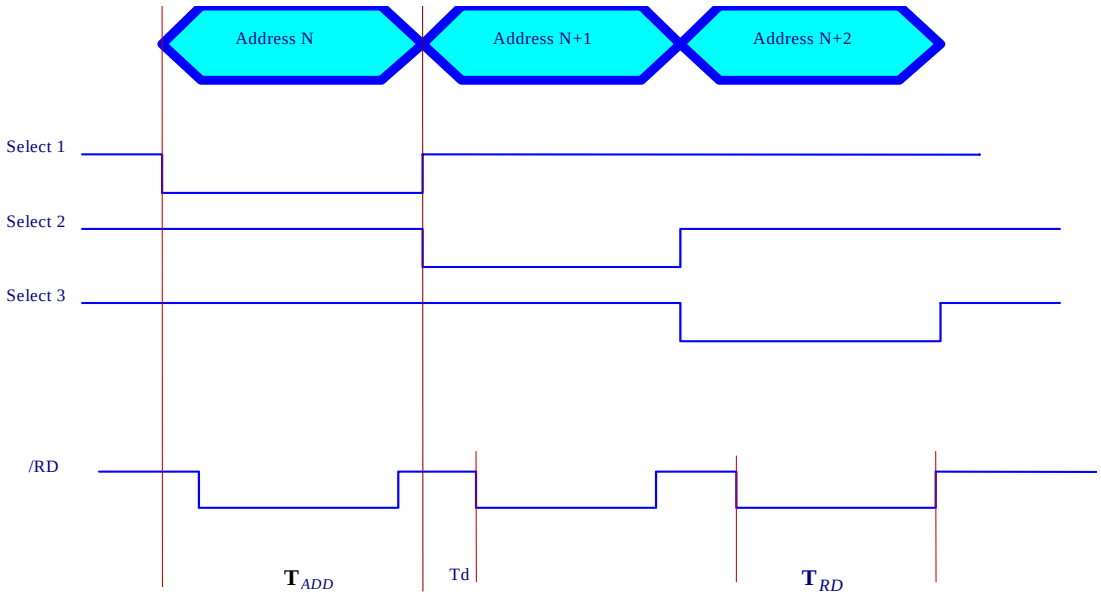


Figure 3a shows the timing of a typical read sequence. Here the sequencer is running through three programmed addresses and sequentially driving each Address onto the Address bus. The Select lines shown above show the response of various devices on the bus decoding their particular address.

The /RD line can be use as both a bus transceiver enable (when it goes low) and a register clock. The application can gate the data from the device at the selected address onto the bus, when the /RD line goes low. The COM Module will latch the data on the bus on the low to high transition of the /RD line.

Assuming a 20 word frame as defined above, and a clock frequency of 50Mhz, the frame rate will be:

50Mhz/ word / 20 words/ frame = 2.5M/frames per sec = 400ns per frame.

$T_{ADD}$ = 400 ns

Assuming that the clock for the FPGA is 50 Mhz -> Td(min) = 1/50Mhz = 20nS

$T_d$ = 20 nS.

$T_{RD}$ = 360 ns.

Doc_info_start
Title: *General Purpose SERDES Communications Protocol Standard*
Document Type: *Report*
Source: *Steward Observatory*
Issued by: *J. Rosato*
Date_of_Issue: *2/16/07*
Revised by:
Date_of_Revision:
Checked by:
Date_of_Check:
Accepted by:
Date_of_Acceptance:
Released by:
Date_of_Release:
File Type: *MS Word*
Local Name:
Category: *675*
Sub-Category: *001*
Assembly: *Instrument Rotator and Cable Chain*
Sub-Assembly:
Part Name:
CAN Designation: *675s001*
Revision: *a*
Doc_info_end